

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Утверждено на заседании
УМС ИАТЭ НИЯУ МИФИ
Протокол №2-8/2024 От 30.08.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Оркестрация и автоматизация в Airflow

Шифр, название дисциплины

01.04.02 «Прикладная математика и информатика»

Шифр, название специальности/направления подготовки

Математическое моделирование и прикладной анализ данных

Название программы магистратуры

магистр

(Квалификация (степень) выпускника)

Форма обучения: очная

г. Обнинск 2024 г.

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 01.04.02 – Прикладная математика и информатика. (квалификация (степень) магистр).

Программу составил:

_____ С.В. Ермаков, доцент, к.ф.-м.н, доцент

Рецензент:

_____ Г.Е. Деев, доцент, к.ф.-м.н, доцент

Программа рассмотрена на заседании ОИКС

(протокол № 5/7 от «30» июля от 2024 г.)

Руководитель направления подготовки 01.04.02
«Прикладная математика и информатика»

_____ Ермаков С.В.

« ____ » _____ 2024 г.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенций	Результаты освоения ООП <i>Содержание компетенций*</i>	Перечень планируемых результатов обучения по дисциплине**
ПК-6	способен к проектированию и разработке наукоемкого программного обеспечения на основе технического задания	З-ПК-6 Знать основные цели и задачи проектирования и разработки наукоемкого программного обеспечения на основе технического задания. У-ПК-6 Уметь разрабатывать наукоемкое программное обеспечение на основе технического задания. В-ПК-6 Владеть навыками разработки и проектирования наукоемкого программного обеспечения на основе технического задания.
ПК-5	способен четко формулировать цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач	З-ПК-5 Знать основные цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач. У-ПК-5 Уметь четко формулировать цели и задачи научно-прикладных проектов, разрабатывать концептуальные и теоретические модели решаемых задач В-ПК-5 Владеть навыками разработки теоретических моделей решаемых задач.

2. Место дисциплины в структуре ООП магистратуры

Дисциплина реализуется в рамках профессионального модуля.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: Docker, Python для анализа данных.

Дисциплина изучается на 1 курсе во 2 семестре.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 3 зачетных единиц (з.е.), 108 академических часа.

3.1. Объем дисциплины по видам учебных занятий (в часах)

	Семестр		
	№ 1	№ 2	Всего

	Количество часов на вид работы:		
Контактная работа обучающихся с преподавателем			
Аудиторные занятия (всего)		64	64
В том числе:			
<i>лекции</i>		32	32
<i>практические занятия</i>		32	32
<i>лабораторные занятия</i>			
Промежуточная аттестация			
В том числе:			
<i>зачет</i>		Зачет с оценкой	Зачет с оценкой
<i>экзамен</i>			
Самостоятельная работа обучающихся (всего)		44	44
В том числе:			
<i>проработка учебного (теоретического) материала</i>		10	10
<i>выполнение индивидуальных заданий</i>		10	10
<i>подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)</i>		10	10
<i>подготовка ко всем видам контрольных испытаний промежуточной аттестации (по окончании семестра)</i>		14	14
<i>Всего (часы):</i>		108	108
<i>Всего (зачетные единицы):</i>		3	3

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела /темы дисциплины	Общая трудоём- кость всего (в часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоёмкость (в часах)				Формы текущего контроля успевае- мости
			Аудиторные учебные занятия			СРО	
			Лек	Сем/Пр	Лаб		
1.		32	32	32	-	44	
1.1.	Знакомство с Apache Airflow	4	2	2		2	
1.2.	1-Понятие «оркестратор» и его	6	2	2	-	2	

	место в ETL						
1.3.	Развертывание Apache Airflow для локальной разработки	6	2	2	-	2	
1.4.	Основные абстракции	6	2	2	-	2	
1.5.	Операторы	6	2	2		2	
1.6.	TaskFlow	6	2	2	-	2	
1.7.	Connection	6	2	2	-	2	
1.8.	Пишем первый `DAG`	6	2	2		2	
1.9.	Архитектура Apache Airflow	6	2	2		2	
1.10	Особенности работы CeleryExecutor	6	2	2	-	2	
1.11.	Работа с Jinja	6	2	2		3	
1.12.	Сенсоры	6	2	2		3	
1.13	Разработка операторов	7	2	2		3	
1.14	Генерация DAG'ов	7	2	2		3	
1.15	Разработка сенсоров	7	2	2		3	
1.16	Финальный проект	7	2	2		9	

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Знакомство с Apache Airflow	зачем нужен Apache Airflow и какие задачи для бизнеса он решает
1.2.	1-Понятие «оркестратор» и его место в ETL	место Apache Airflow в ETL-процессах компании
1.3.	Развертывание Apache Airflow для локальной разработки	как разворачивать Apache Airflow с Celery Executor через docker-compose
1.4.	Основные абстракции	понятия `DAG` и `DagRun` и их отличия понятиями `Task` и `TaskInstance` и их отличия
1.5.	Операторы	для чего нужны `PythonOperator` и `BashOperator`
1.6.	TaskFlow	как выстраивать зависимости между `Task` 2 способами
1.7.	Connection	как создать новый `Connection` для чего нужны `Hooks` Познакомились с концепцией `Providers`

1.8.	Пишем первый `DAG`	Пишем первый `DAG`
1.9.	Архитектура Apache Airflow	Архитектура Apache Airflow
1.10	Особенности работы CeleryExecutor	что делает Executor и какие виды бывают, их отличия
1.11.	Работа с Jinja	Реализация идемпотентного DAG'а через `{{ ds }}`» концепция идемпотентности DAG и почему она важна основные встроенные макросы Jinja смысл `template_fields` Написание кастомных макросов для Jinja
1.12	Сенсоры	особенности сенсоров и их отличие от операторов
1.13	Разработка операторов	архитектура операторов в Apache Airflow BaseOperator, паттерн `Command`, метод `execute`
1.14	Генерация DAG'ов	Реализация батчевой выгрузки данных из БД в CSV» (4 DAG)
1.15	Разработка сенсоров	трешхолд SQL-сенсор
1.16	Финальный проект	финальный проект

Практические/семинарские занятия

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Знакомство с Apache Airflow	зачем нужен Apache Airflow и какие задачи для бизнеса он решает
1.2.	Понятие «оркестратор» и его место в ETL	место Apache Airflow в ETL-процессах компании
1.3.	Развертывание Apache Airflow для локальной разработки	как разворачивать Apache Airflow с Celery Executor через docker-compose
1.4.	Основные абстракции	понятия `DAG` и `DagRun` и их отличия понятиями `Task` и `TaskInstance` и их отличия
1.5.	Операторы	для чего нужны `PythonOperator` и `BashOperator`
1.6.	TaskFlow	как выстраивать зависимости между `Task` 2 способами
1.7.	Connection	как создать новый `Connection` для чего нужны `Hooks` Познакомились с концепцией `Providers`

1.8.	Пишем первый `DAG`	Пишем первый `DAG`
1.9.	Архитектура Apache Airflow	Архитектура Apache Airflow
1.10	Особенности работы CeleryExecutor	что делает Executor и какие виды бывают, их отличия
1.11.	Работа с Jinja	Реализация идемпотентного DAG'a через `{{ ds }}`» концепция идемпотентности DAG и почему она важна основные встроенные макросы Jinja смысл `template_fields` Написание кастомных макросов для Jinja
1.12	Сенсоры	особенности сенсоров и их отличие от операторов
1.13	Разработка операторов	архитектура операторов в Apache Airflow BaseOperator, паттерн `Command`, метод `execute`
1.14	Генерация DAG'ов	Реализация батчевой выгрузки данных из БД в CSV» (4 DAG)
1.15	Разработка сенсоров	трешхолд SQL-сенсор
1.16	Финальный проект	финальный проект

Лабораторные занятия

Не предусмотрены.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

В качестве учебно-методических материалов используется рекомендованная литература.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
1.3	Развертывание Apache Airflow для локальной разработки	ПК-5	Контрольная работа № 1
1.9.	Архитектура Apache Airflow	ПК-5	Контрольная работа № 1
1.11.	Работа с Jinja	ПК-6	Контрольная работа № 2
1.15.	Разработка сенсоров	ПК-5	Контрольная работа № 2

6.2. Типовые контрольные задания или иные материалы

6.2.1. Зачет с оценкой

В билете два теоретических вопроса: один по аналитическим методам оптимизации, второй – по численным методам оптимизации.

Теоретические вопросы билета:

В билете два теоретических вопроса и одного практического.

Теоретические вопросы билета:

1. Что такое оркестратор в контексте ETL-процессов и какова его роль?
2. В чем заключается основное отличие между оркестратором задач и традиционными средствами планирования задач?
3. Опишите процесс развертывания Apache Airflow для локальной разработки. Какие шаги необходимо выполнить?
4. Каковы основные абстракции в Apache Airflow, и как они взаимодействуют друг с другом?
5. Что такое операторы в Apache Airflow и как они используются для реализации задач?
6. В чем заключается концепция TaskFlow в Apache Airflow и как она улучшает разработку рабочих процессов?
7. Как настроить и использовать Connection в Apache Airflow для подключения к внешним системам и базам данных?
8. Опишите архитектуру Apache Airflow и основные компоненты системы.
9. Как Airflow поддерживает параллельное выполнение задач, и какие механизмы для этого используются?
10. Какие типы операторов существуют в Apache Airflow и для каких задач они предназначены?
11. Как выстроить зависимости между задачами в DAG с помощью Apache Airflow?
12. В чем отличие между статическими и динамическими DAG в Apache Airflow?
13. Какие способы управления ошибками существуют в Apache Airflow для задач, которые не могут быть выполнены?
14. Как можно масштабировать Apache Airflow для работы с большими объемами данных или высокой нагрузкой?

Каковы основные подходы к мониторингу и логированию задач в Apache Airflow?

6.2.2. Контрольная работа № 1

Описание задачи:

Создайте и разверните ETL-пайплайн в Apache Airflow, который выполняет следующие задачи:

Извлечение данных:

Скачайте CSV-файл с открытыми данными (например, список пользователей) с использованием BashOperator.

Трансформация данных:

Преобразуйте данные с помощью PythonOperator: приведите все имена в верхний регистр и отфильтруйте пользователей, чей возраст меньше 18 лет.

Загрузка данных:

Сохраните преобразованные данные в новый CSV-файл или загрузите их в базу данных SQLite

с использованием PythonOperator.

Требования:

Разработать DAG, который выполняет все три задачи последовательно.

Пайплайн должен быть развернут локально с использованием Apache Airflow.

Обработать возможные ошибки, например, если файл не удастся скачать.

6.2.2. Контрольная работа № 2

Задача: Создание и настройка DAG для обработки данных

Описание задачи:

Создайте простой DAG в Apache Airflow, который выполняет следующие операции:

Извлечение данных:

Загрузите данные с открытого API (например, данные о погоде) с использованием PythonOperator. Выведите полученные данные в консоль.

Трансформация данных:

Преобразуйте данные (например, извлеките только температуру и описание погоды) с помощью PythonOperator.

Загрузка данных:

Сохраните преобразованные данные в текстовый файл с использованием PythonOperator.

Требования:

Создайте DAG, который выполняет все три задачи последовательно.

Пайплайн должен быть настроен для локальной разработки.

Добавьте обработку ошибок (например, уведомление в случае неудачной загрузки данных).

б) критерии оценивания компетенций (результатов) – правильная работа кода программы, понимание алгоритма метода оптимизации, умение вывести необходимые для алгоритма формулы.

в) описание шкалы оценивания:

Каждая задача оценивается по шкале от 0 до 10 баллов.

Контрольная работа считается выполненной успешно при суммарной оценке не ниже 18 баллов.

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Форма аттестации	Наименование оценочного средства	Баллы
Зачет (100 баллов)	Контрольная работа № 1	30
	Контрольная работа № 2	30
	Ответы на билет	40

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

1. James D. Miller – Apache Airflow: The Hands-On Guide – 2020 – 360 страниц
2. Marcelo Leal – Data Pipelines with Apache Airflow – 2020 – 348 страниц
3. Cesar A. L. de A. – Learning Apache Airflow – 2020 – 220 страниц

б) дополнительная учебная литература:

1. Patrick S. McDaniel – Mastering Apache Airflow – 2021 – 400 страниц
2. Tarek S. S. J. S. – Apache Airflow for Data Engineering – 2020 – 250 страниц

8. Перечень ресурсов* информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины

-

9. Методические указания для обучающихся по освоению дисциплины

Вид учебного занятия	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.
Практические занятия	Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Работа с конспектом лекций, просмотр рекомендуемой литературы. Изучение выбранной предметной области на примерах решения задач семинарских занятий, индивидуальных домашних заданий.
Курсовая работа	Не предусмотрена
Контрольная работа	Ознакомиться с основной и дополнительной литературой, включая справочные издания, зарубежные источники, основополагающие термины. Попрактиковаться в решении аналогичных домашних задач по всем темам контрольных работ.
Лабораторная работа	Не предусмотрена.
Подготовка к зачету	При подготовке к зачету необходимо ориентироваться на конспекты лекций и рекомендуемую литературу.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Видеопроектор, компьютер, издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Часов в интерактивной форме – 8.

В ходе практических занятий происходит публичное обсуждение каждой решаемой задачи. При этом студенты высказывают свои мнения по выбору наиболее простого способа поиска оптимального решения.

После решения домашних работ на консультациях проводится разбор допущенных студентами ошибок.

12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Некоторые темы изучаются студентами самостоятельно. Для изучения используется приведённая в списке основная и дополнительная литература. Контроль освоения материала осуществляется при проверке контрольных работ, домашнего задания и на зачете.

№	Тема и часть, изучаемая (осваиваемая) самостоятельно
1.1	Основы ETL (Extract, Transform, Load) процессов
1.2	Параллельное и асинхронное выполнение задач
1.3	Интеграция Apache Airflow с другими инструментами для работы с данными
1.4	Использование Apache Airflow для автоматизации Machine Learning процессов
1.5	CI/CD для Apache Airflow
1.6	Мониторинг и логирование процессов в Airflow
1.7	Распределенные системы и их использование для обработки больших данных
1.8	Cloud-native подходы для развертывания Apache Airflow (Kubernetes, GCP, AWS)
1.9	Использование и интеграция с базами данных (например, PostgreSQL, MySQL) в Airflow
1.10	Обзор других систем оркестрации задач: Apache NiFi, Celery, Kubeflow

Вопросы и задания для самоконтроля по всем темам:

1. Что такое оркестратор в контексте ETL и какую роль он играет в процессе обработки данных?
2. Какие преимущества использования оркестраторов данных, таких как Apache Airflow, в сравнении с традиционными методами обработки задач?
3. Как настроить Apache Airflow для локальной разработки? Какие основные шаги необходимо выполнить при установке?
4. Что такое DAG в Apache Airflow и какие основные компоненты входят в его структуру?

5. Что такое операторы в Apache Airflow и как они используются для выполнения различных задач в DAG?
6. Как работает TaskFlow в Airflow и чем он отличается от стандартного подхода к созданию зависимостей между задачами?
7. Как настроить Connection в Apache Airflow для подключения к внешним сервисам и базам данных? Какие параметры конфигурации необходимо указать?
8. Опишите основные абстракции Apache Airflow, такие как задачи (Tasks), операторы (Operators) и DAG.
9. Как управляются зависимости между задачами в Apache Airflow и как можно настроить выполнение задач в определенной последовательности?
10. Чем отличается PythonOperator от BashOperator в Apache Airflow, и когда стоит использовать один из них?
11. Как организовать параллельное выполнение задач в Airflow и какие настройки необходимы для этого?
12. Какие типы операторов существуют в Apache Airflow и как выбрать подходящий для выполнения конкретной задачи?
13. Как настроить мониторинг выполнения задач и обработку ошибок в Apache Airflow?
14. Что такое архитектура Apache Airflow и каковы основные компоненты системы (например, Web Server, Scheduler, Worker)?
15. Как Airflow поддерживает масштабирование и как можно настроить его для работы с большими объемами данных?

12.3. Краткий терминологический словарь

DAG (Directed Acyclic Graph)	Направленный ациклический граф, который представляет собой структуру для оркестрации задач в Apache Airflow. DAG описывает зависимость задач и их последовательность выполнения.
Task Instance	Конкретная реализация задачи, которая включает в себя информацию о том, когда задача была выполнена, ее состояние и результат.
Scheduler	Компонент Apache Airflow, который управляет запуском задач в соответствии с расписанием, указанным в DAG. Он следит за зависимостями и состоянием задач, чтобы правильно организовать их выполнение.